

FIG. 1

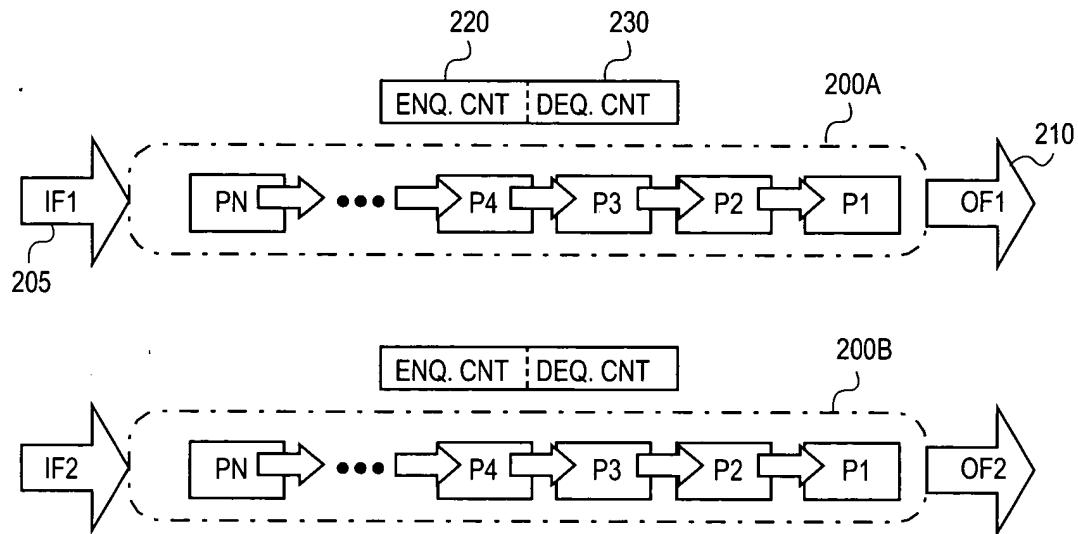


FIG. 2

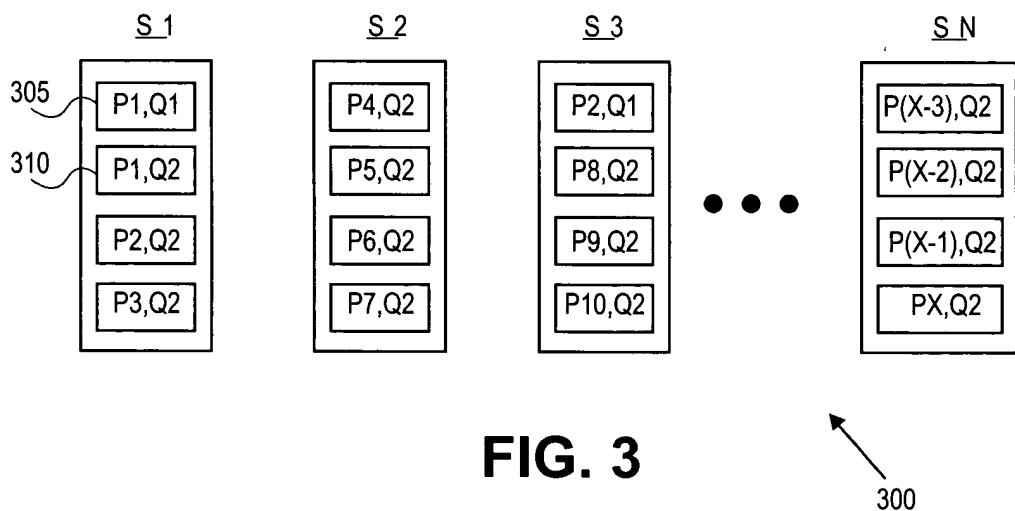


FIG. 3

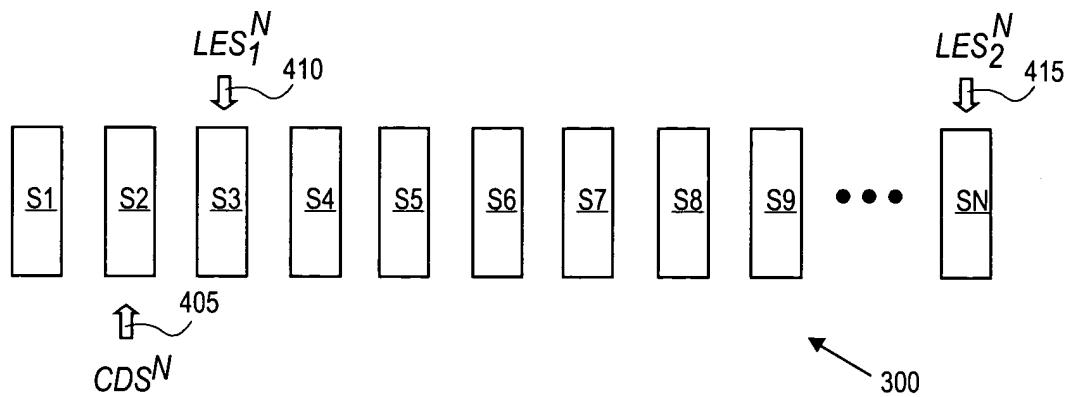


FIG. 4

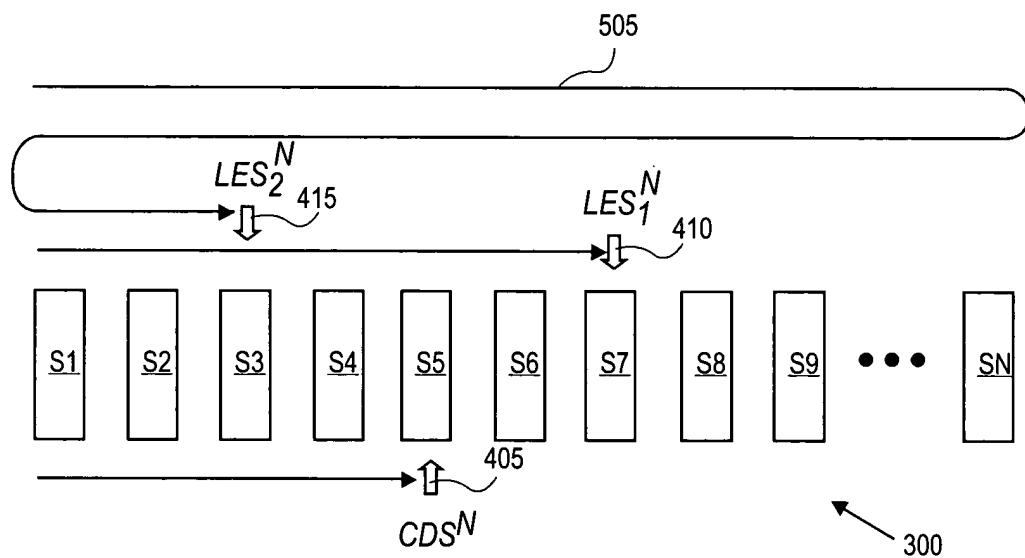


FIG. 5

```

610
enqueue( queue i)
{
    if (enqueue_count == dequeue_count) // check #1
    {
        LESiN = CDSiN;           // queue is empty
    }
    else if (((CDSiN - LESiN) mod N) < M) // check #2
    {
        LESiN = CDSiN;           // queue is empty and dequeue count is
        // lagging
    }

    LESiN = (LESiN + j) mod N;           // calculate where to enqueue the packet
    // value j < M depends on queuing scheme
    // Note: LESiN may increase, CDSiN unchanged
    if (((CDRN - LSRiN) mod N) < M) // check #3
    {
        Drop packet           // queue has overflowed
        LESiN = (LESiN - j) mod N; // reset LESi to old value
    }
    else
    {
        Enqueue the packet
    }
}
605
while (1)
{
    enqueue(i);           // call enqueue routine for queue
    // value LESi may increase
    Perform some dequeues
    Perform enqueues on other queues
    CDSN = (CDSN + k) mod N; // where k >= 0, k depends on how many rounds
    // have completed dequeues
    // Note: LESiN unchanged, CDSN may increase
}

```

FIG. 6

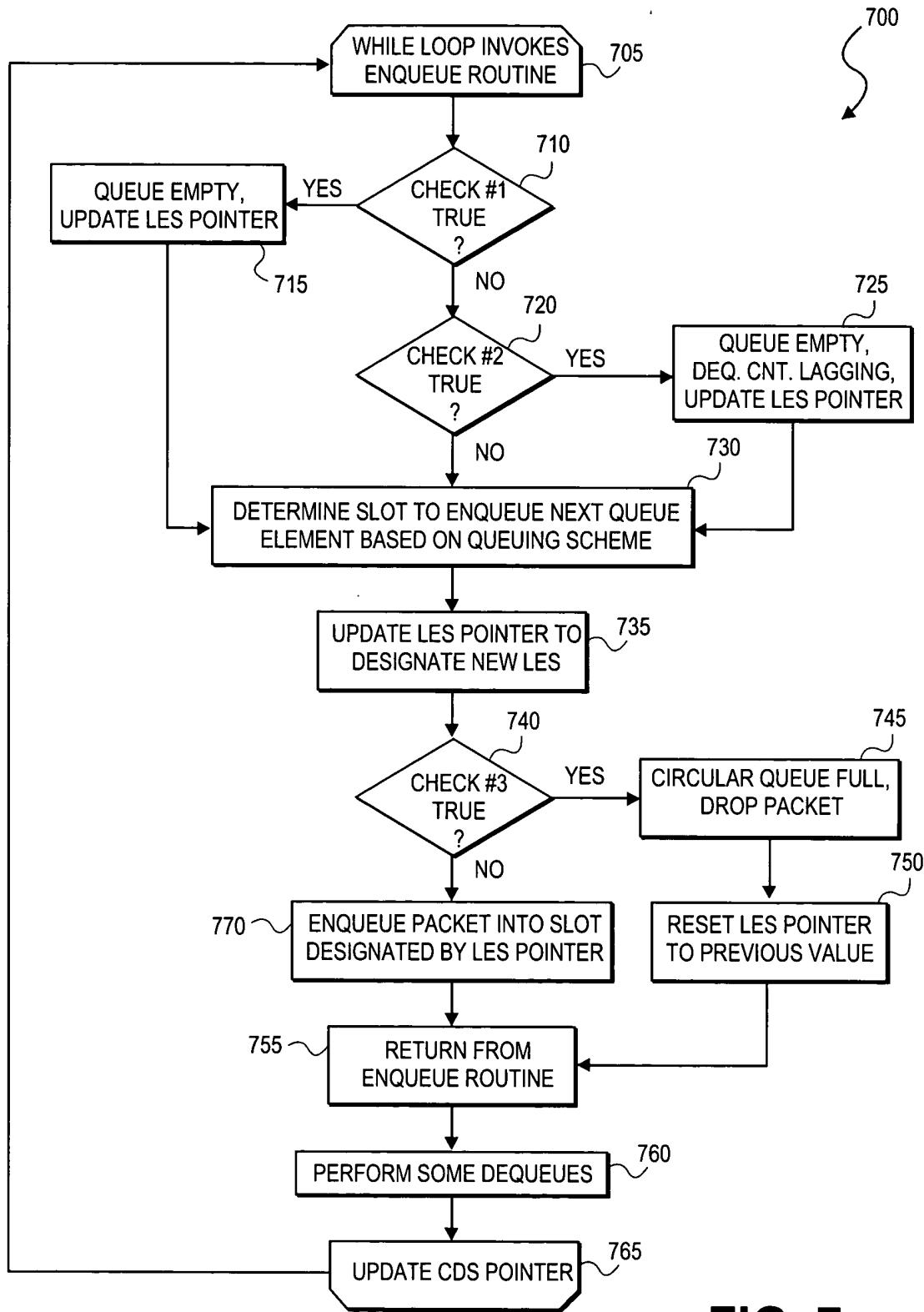


FIG. 7

CHECK#1: $ENQ_CNT == DEQ_CNT$

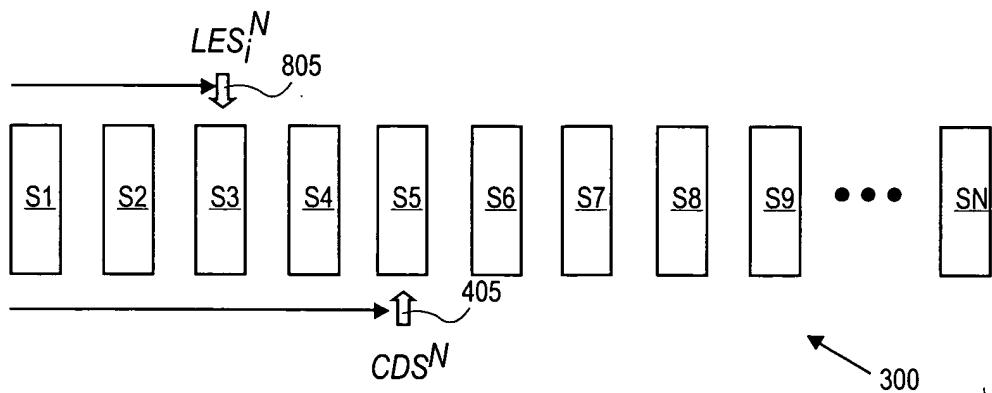


FIG. 8

CHECK #2: $((CDS^N - LES_i^N) \bmod N) < M$

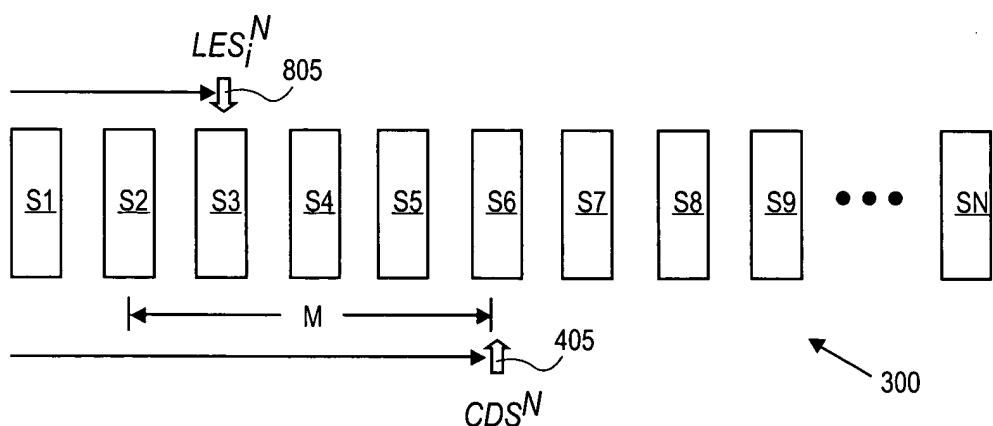


FIG. 9

CHECK #3: $((CDS^N - LES_i^N) \bmod N) < M$

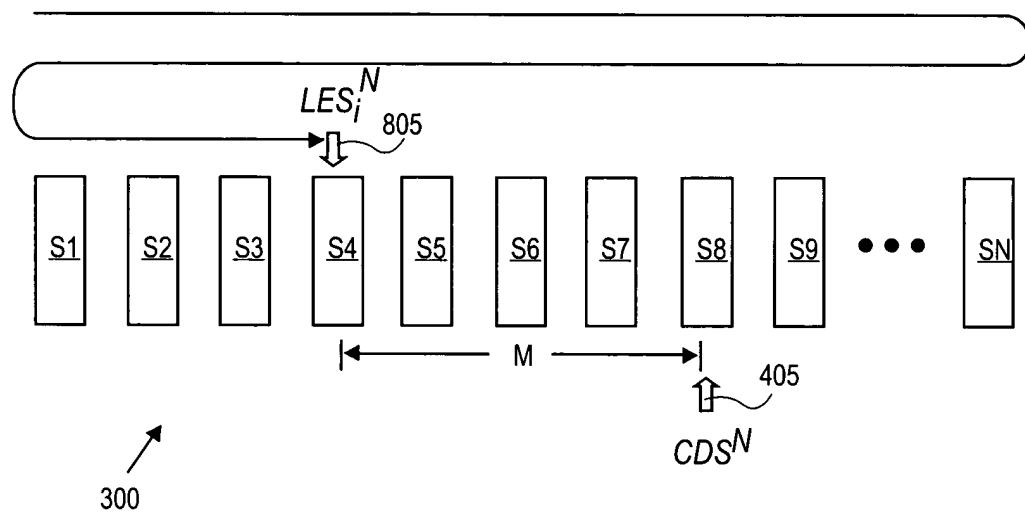


FIG. 10

FIG. 11

